Self-supervised Spatial Reasoning on Multi-View Line Drawings

Siyuan Xiang[*‡] Anbang Yang[*‡] Yanfei Xue[‡] Yaoqing Yang[§] Chen Feng[†‡]
[‡]New York University Tandon School of Engineering [§]University of California, Berkeley
https://ai4ce.github.io/Self-Supervised-SPARE3D/

Appendix

A. Supervised Learning Exploration

We have explored three factors that might affect the performance of supervised learning based methods: (1) the quality of data used for training, (2) the network's capacity (width and depth), and (3) the network's structure. We show the influence of factor 1 in the paper, and we will show the influences of factor 2 and 3 in the supplementary. Since task *I2P* and *P2I* both belong to *camera pose reasoning* task, we only test the factor influence on *I2P* for the sake of limited computational resources.

A.1. Network's Capacity Exploration

Network depth for T2I and I2P tasks. The depth of the network represents the number of layers of the VGG family backbone networks. We use VGG-13, VGG-19 based backbone to represent three different depths of the network.

Network width for T2I and I2P tasks. Table 1 shows the detailed network width for all VGG-16 based backbone network with different widths, for *T2I* and *I2P* tasks.

For task *T2I*, decreasing the network's width does not hurt the network's performance, although strangely, increasing the network's width leads to a decrease in the testing accuracy (Figure 1 top-left). For the depth control experiments, we find almost no differences among the three selected network depth values (Figure 1 top-right). For task *I2P*, neither the width nor the depth of the network can significantly improve the network's performance(Figure 1 bottom).

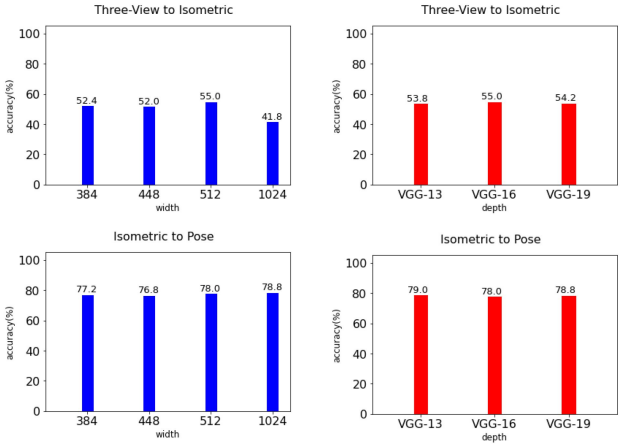

Figure 1. **Network capacity (width and depth) vs. test accuracy**. The results show that naively increasing the network capacity cannot improve the network performance on *T2I* and *I2P* tasks.

	384	448	512	1024
$1-2$	48	56	64	128
$3-4$	96	112	128	256
$5-7$	192	224	256	512
$8-13$	384	448	512	1024

Table 1. **Network width for VGG-16 based backbone network.** 384, 448, 512, 1024 are the width of the modified VGG-16 backbone network. $1-2$, $3-4$, $5-7$, $8-13$ means convolutional layer $1-2$, $3-4$, $5-7$, $8-13$ respectively. The number in each cell means the width of that convolutional layer (the number of channels).

A.1.1 Network Structure Exploration

P2I network structure. As mentioned in our paper, to make a fair comparison, we modify the supervised baseline method for the *P2I* task so that it has the same network structure for feature extraction as our self-supervised network. In this section, we introduce the details of the modified network structure for the *P2I* task.

Each question in *P2I* task contains the 3-view(front, right, top view) line drawings and one given pose out of eight views. We note these three drawings as F, R, T, and the given pose as P_m, ($m \in \{1,2,3,4,5,6,7,8\}$). The answers provide four candidate drawings. These candidate drawings are isometric view line drawings rendered from four views, which are randomly selected from eight views designed in SPARE3D dataset. We note the four candidate answers as I_1, I_2, I_3, I_4. For the *early fusion method*, we concatenate F, R, T and one of the isometric drawings I_i,

[*]Equal contribution.
[†]The corresponding author is Chen Feng cfeng@nyu.edu.

($i \in \{1,2,3,4\}$) to form a $12-channel$ composite image I_{c_i}, ($i \in \{1,2,3,4\}$). Then, we send I_{c_i} to a VGG-based classifier: $g_\theta : \mathbb{R}^{12\times\ H\times W} \rightarrow \mathbb{R}^8$, where θ represents the parameters in the network. The 8 number codeword represents the probability of the composed image I_{c_i} belonging to eight coded views. Then, we pick the number of the codeword corresponding to view the P_m, note as $\hat{p}_{m_i}$, ($i \in \{1,2,3,4\}$). The ground truth is set to be 1 if the candidate isometric drawing is rendered from view P_m, and otherwise 0. With the provided ground truth p_m, we can compute the BCE loss to train the neural network, which is: $\frac{1}{4}\sum_{k=1}^{4} BCE(\hat{p}_{m_k}, p_{m_k})$.

Network structure for all tasks. As we mentioned in the paper, we explore many variants of the baseline network structure, to see if a variant will affect the network's performance on the task. Because of the limitation of our computational resources, we explore the variants on the task *I2P*. Here, we list out all the six variants we tried and the results in Table 2. Among them, two variants have a consistent impact on the tasks, which are (1) whether using pre-trained parameters from ImageNet, (2) using *early fusion* or *late fusion* for image feature extraction.

Early fusion vs. *late fusion*. In the original SPARE3D paper, the baseline backbone network treats all the input images (front, right, top view drawings, and one isometric view drawing from the candidate answers) as a whole, and it concatenates those images before sending them to the first convolutional layer. We call this way of feeding multi-view line drawings to a network as the *early fusion*. In contrast, we design a network that takes the three-view drawings and the isometric view drawing as separate inputs, which means the input drawings are sent to a convolutional network that shares the *same architecture* yet has *separate network parameters*. We name this way of separately handling the input as the *late fusion* since the extracted image features are concatenated later. Other network structures are kept the same as the baseline method in SPARE3D.

Pre-training vs. *No pre-training*. As in many other research works, we find using ImageNet pre-trained parameters for the backbone VGG network has obvious influnece on our tasks.

Next, we will focus on the remaining four variants that do not have obvious influence on our tasks: (1)"no pooling", (2)"no dropout", (3)"share weight", and (4)"separate fc" respectively. "no pooling" means we discard all the adaptive average pooling layer in the VGG-16 backbone. "no dropout" means we delete all the dropout layers in the VGG-16 backbone. "share weight" means for the *late fusion method*, all the VGG-16 backbone use the *same architecture* and with *same parameters*. "separate fc" means for the *late fusion method*, the front, right, top view drawings are first fed into the VGG-16 backbone based network: $g_\phi : \mathbb{R}^{3\times\ H\times W} \rightarrow \mathbb{R}^{18432}$. We note the image features as c_f, c_r, c_t separately. Then we concatenate the three codewords to form a code word, and maps it via an MLP: $g_\psi : \mathbb{R}^{55296} \rightarrow \mathbb{R}^{18432}$. For the isometric drawing, we send it to the VGG-16 backbone based network: $g_\phi : \mathbb{R}^{1\times\ H\times W} \rightarrow \mathbb{R}^{18432}$. Finally, we concatenate the two codewords generated from 3-view drawings and isometric drawing as the feature vector for the classification. Other parts of the network are the same as not using the "separate fc" structure.

Table 2 reveals that "no pooling", "no dropout", "share weight", "separate fc" has no obvious and consistent impact on the network's performance. Since rows with an odd number as index differ from the rows with even numbers in "pre-train", each time we will compare two odd rows, which are not using "pre-train". The same conclusion can be drawn if we compare two even rows each time.

We can compare the 1 row with the 3 row, and we can find that "no pooling" does not obviously affect the results. If we compare the 1 row with the 5 row, we can find "no dropout" also cannot help the network perform better. Comparing the 1 row with the 7 row, we have the conclusion that using both "no pooling" and "no dropout" could not improve the network's classification accuracy.

For $9, 11, 13$ rows, we use the *late fusion method*. Based on this method, we vary the network's structure of the remaining four variants. We also find these four variants do not have a significant influence on the network for *late fusion method*. For 9 row, "no pooling" and "no dropout" seem not to impact on the classification results; for 11 row, "no pooling", "no dropout", and "separate fc" does not work; for 13 row, all the four variants cannot help improve the performance.

Therefore, we conclude that except for the two factors we mentioned in the previous section, the other four factors do not have an obvious impact on the final classification results on the *I2P* task.

B. Twenty Camera Poses for Extension Tasks

As we mentioned in our paper, we extend the *I2P* and *P2I* tasks to *extension I2P* and *extension P2I* using twelve more camera poses. We show all the twenty camera poses in Figure 2.

C. Additional Attention Maps for *T2I*

We provide more visualization results of attention maps to compare our method with supervised learning method, as in Figure 3.

index	pre-train	late fusion	no pooling	no dropout	share weight	separate fc	max	avg
1	n	n	n	n	n	n	76.8	72.5
2	y	n	n	n	n	n	80.4	78.9
3	n	n	y	n	n	n	76.4	71.1
4	y	n	y	n	n	n	79.4	79.1
5	n	n	n	y	n	n	77.8	69.1
6	y	n	n	y	n	n	81.2	80.9
7	n	n	y	y	n	n	70.4	66.8
8	y	n	y	y	n	n	80.8	79.1
9	n	y	y	y	n	n	78.8	77.4
10	y	y	y	y	n	n	**86.4**	84.1
11	n	y	y	y	n	y	80.0	76.0
12	y	y	y	y	n	y	85.4	84.3
13	n	y	y	y	y	y	75.6	74.2
14	y	y	y	y	y	y	85.4	84.9

Table 2. **Network architecture vs. performance on *I2P* task.** The backbone used is VGG-16. The rows in the green background represent the networks are not initialized with the Imagenet pre-trained parameter, while the rows in the white background are initialized with the parameters. Every two rows (odd row and even row) can be compared to see the influence of using pre-trained parameters or not. We provide the max and average results for each type of network based on seven times of implementation.

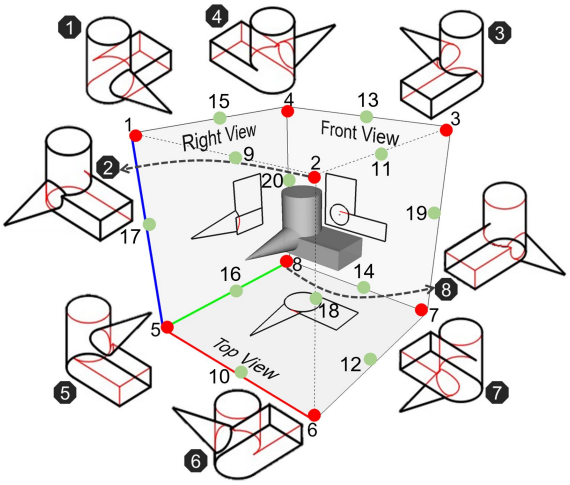

Figure 2. **Twenty camera poses for *extension I2P* and *extension P2I* tasks.**

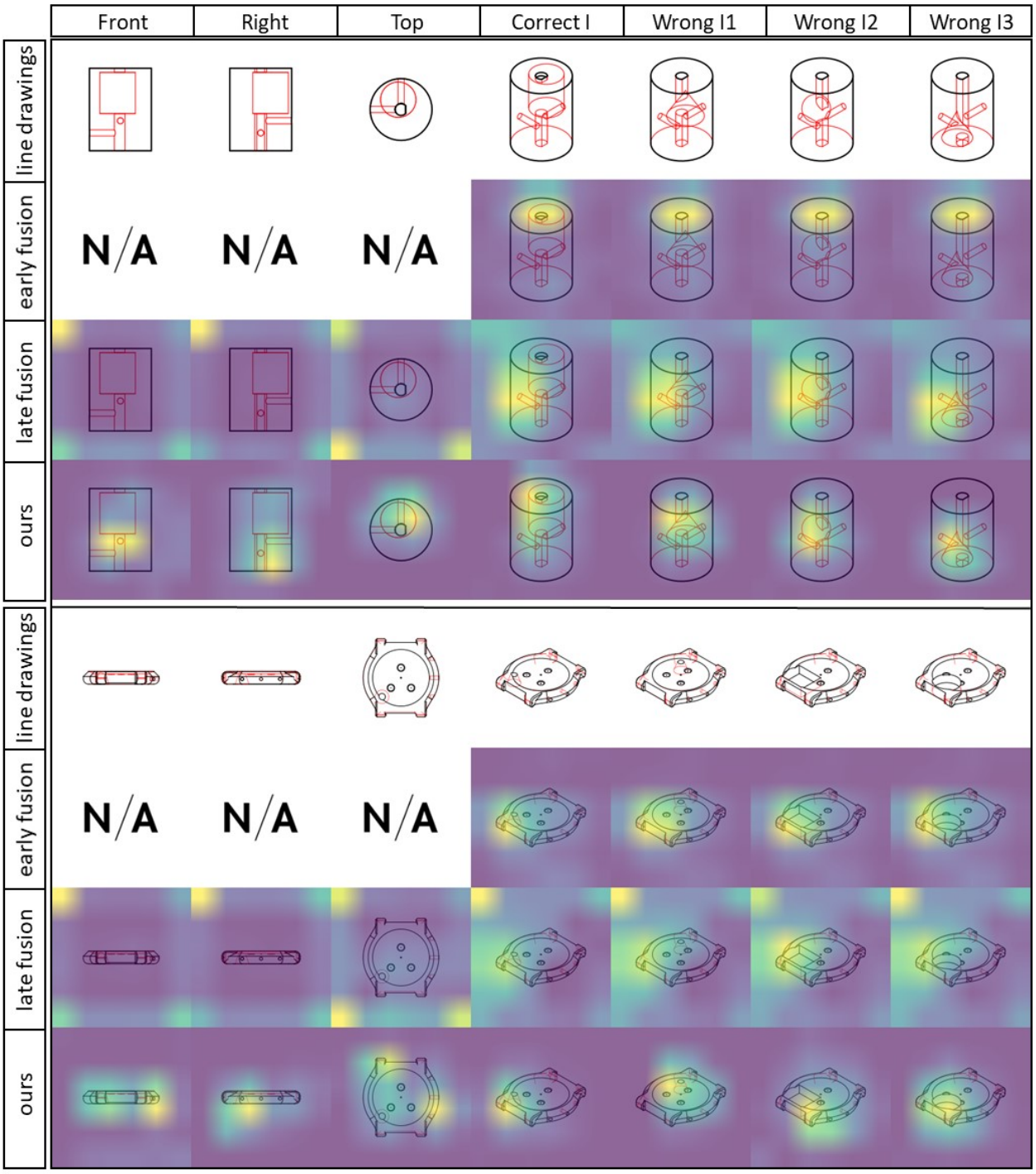

Figure 3. **Attention maps for SL vs. SSL method in *T2I* task.**